

The 6<sup>th</sup> International Conference on Modelling of mechanical and mechatronic systems

## CAD of cascade controllers for DC drives using genetic algorithm methods

Godem Ali M. Ismeal\*, Karol Kyslan, Viliam Fedák

*Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic*

---

### Abstract

PID controller presents the most widely used controller in industrial applications. It needs efficient methods to set-up and tune parameters of its three components based on the properties of the plant and its parameters. In this paper the cascade controller for a DC drive (consisting of the speed and current controllers in series) is designed and tuned using genetic algorithm technique with different objective functions. The method of designing two PID controllers connected in cascade by the genetic algorithm technique was applied to a DC drive equipped by current and speed controllers. The aim of genetic algorithm control method consists in finding the best set of solutions for the controller parameters that give desired dynamic and static performance of the system. By using MATLAB/Simulink software a graphical user interface (GUI) was developed for efficiently, user friendly, and comfortable calculation and tuning the parameters of both controllers in cascade and for visualizing results.

© 2014 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license  
(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of organizing committee of the Modelling of Mechanical and Mechatronic Systems MMaMS 2014

**Keywords:** controller design; PID controller; DC motor; genetic algorithm method; graphical user interface

---

### 1. Introduction

DC motors are still important machines in many industrial applications such as steel rolling mills, electric vehicles and trams, robotic manipulators [1, 2], etc. Classical methods of parameters calculation of the controlled drive with the DC motor are based on knowledge of its mathematical model [3], including model of the driven machine. The

---

\* Corresponding author. Tel.: +421-908-510-850; fax: +421-633-0115  
E-mail address: [ng2008ng@yahoo.com](mailto:ng2008ng@yahoo.com)

mathematical model itself cannot ensure a correct behaviour without correct parameters in the model, so knowledge about values of the system parameters is basic. There are many models to represent the DC motors behaviour with a good accuracy [4, 5]. DC motor speed and current are usually controlled by the speed- and current controllers connected in cascade. In this type of control there are two control loops, inner loop with current controller for controlling the current and outer loop with speed controller for controlling the speed.

The PID controllers are widely used in industrial control applications due to their simple structure, comprehensible control algorithms and low costs. Although many advanced control techniques have been proposed to improve system performance, the conventional PID controllers still operates the majority of the control system in the world. In cascade control of the drives, proportional-integral-derivative (PID) controllers are used to eliminate an overshoot in the current loop and/or an overshoot in speed loop, if required.

The conventional PID controller can be tuned by many classical methods such Ziegler-Nichols method, etc. and also an algorithm for hand-tuning of the controller terms is well known. On the other hand it can be tuned by soft computing methods such as artificial neural network (ANN), fuzzy logic (FL), and genetic algorithm (GA) ones.

Two main problems encountered in motor control are the time-varying nature of motor parameters under operating conditions and existence of noise in system loop. Analysis and control of complex, nonlinear and/or time-varying systems is a challenging task using conventional methods because of uncertainties. The above problems can be solved by using genetic algorithm technique for tuning the PID controller that is presented below.

## 2. Mathematical model of the plant (the drive with DC motor)

The DC motor is described by known set of linear differential and algebraic equations from which we derive the block diagram, shown in Fig. 1.

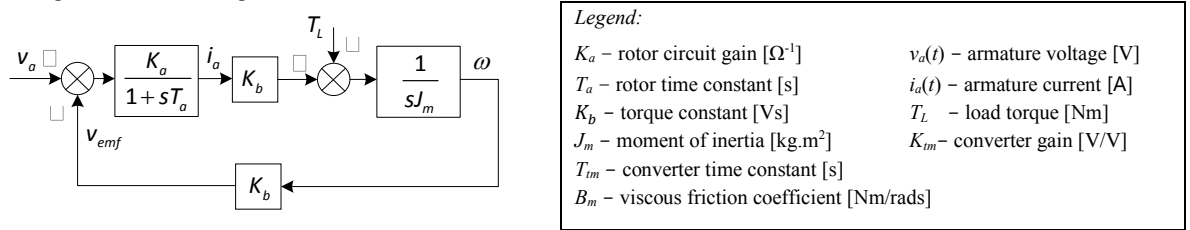


Fig. 1 DC motor block diagram with the legend to explain used constants and variables.

It is easy to derive the transfer function of the motor between angular speed  $\omega(s)$  and armature voltage  $V_a(s)$ :

$$G(s) = \frac{\omega(s)}{v_a(s)} = \frac{K_a K_b}{K_a K_b^2 + sJ_m + s^2 J_m T_a} \quad (1)$$

The DC drive is usually controlled by the current controller having the transfer function  $F_{rc}(s)$  and the speed controller  $F_{rw}(s)$  that are connected in the cascade (Fig. 2).

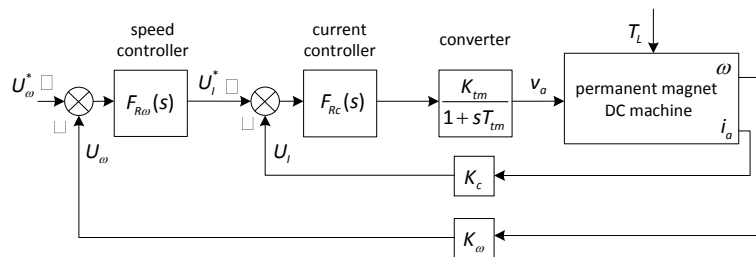


Fig. 2 Block diagram of the drive with DC motor and cascade controllers.

### 3. Genetic algorithm method

Genetic Algorithms were originally developed by John Holland (1975). Genetic Algorithms are a stochastic global search method that mimics the process of natural evolution [6 - 8]. Here is a brief description of how the GA works.

The working of genetic algorithm starts with a random population. A random population is usually consisting of between 20-100 individuals and usually represented by a real-valued number or a binary string called a chromosome. GA operation basically depends on the Schema theorem. The genetic algorithm has three main stages as follows *reproduction, crossover and mutation*.

During the *reproduction* phase the fitness value of each chromosome is assessed. This value is used in the selection process to provide bias towards fitter individuals. Just like in natural evolution, a fit chromosome has a higher probability of being selected for reproduction. This continues until the selection criterion has been met. All selection methods are based on the same principal that is giving fitter chromosomes a larger probability of selection. Four common methods for selection are *roulette wheel selection, stochastic universal sampling, normalized geometric selection, and tournament selection*.

Due to the complexities of the other methods, the *Roulette Wheel method* is preferred in this work. Once the selection process is complete, the crossover algorithm is initiated. The crossover probability indicates how often the crossover is performed. A probability of 0 % means that the offspring will be exact replicas of their *parents* and a probability of 100 % means that each generation will be composed of entirely new offspring. There are a number of crossover techniques such as *single point crossover, multi-point crossover, and uniform crossover*.

Procedure for Genetic Algorithm method application is as follows [9, 10]:

1) *Population Size*

The first step in the implementation of the GA is to establish an initial population. The initial population typically consists of 20–100 individual strings which in themselves represent potential solutions in the search for global optima. There is not any derived method or rule how to choose the best population size - it is based on trial and error.

2) *Reproduction*

During the reproduction phase the fitness value of each chromosome is assessed. This value is used in the selection process to provide bias towards fitter individuals. Just like in natural evolution, a fit chromosome has a higher probability of being selected for reproduction. There are a number of selection methods available and all selection methods are based on the same principal that is giving fitter chromosomes a larger probability of selection. Four common methods for selection are: *Roulette Wheel selection, Stochastic Universal sampling, Normalized geometric selection, and Tournament selection*.

3) *Crossover*

Once the selection is finished the crossover is performed. Individuals are paired for mating and by mixing their strings new individuals are created. Crossover is another process that involves exchange of genetic materials between two parent chromosomes to make child chromosome. There are many types of crossover such as *single-point crossover, two-points crossover, and Uniform crossover*.

4) *Mutation*

In natural evolution, mutation is a random process that causes one part of a gene to change which results in an entirely new genetic structure. Mutation in genetic algorithms, like its equivalent in nature, occurs with a low probability – typically in the range from 0,01 to 0,001.

#### 5) Application of genetic algorithm method for tuning PID controller parameters

The PID controller consisting of three terms - the proportional one with the gain  $K_P$ , the integral  $K_I$  and derivative term  $K_D$ , is described by the equation:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e \quad (2)$$

where  $u(t)$  is output of the controller (the controlling signal sent to the system) and  $e(t)$  is the tracking error

$e(t) = r(t) - y(t)$ , with  $y(t)$  as the measured output and  $r(t)$  – the desired input (reference value).

In the next we will apply the GA method to tune current and speed controllers connected in cascade that is verified by simulation. The procedure of tuning is similar to the classical method of tuning the controllers by Absolute Value Optimum Criterion (the current controller) and Symmetric Optimum Method (the speed controller) is as follows:

- 1) The current controller of the PID type is tuned for the subsystem containing the DC motor and converter.
- 2) Values the controller parameters are implemented into a Simulink model.
- 3) Afterwards, the speed controller is tuned for the subsystem with current controller.

The main objective of the GA method controller design is to find the optimal values of the parameters of PID controllers in order to get a better transient response of the system. The chromosome is formed by three values that correspond to three gains of the controller terms to be adjusted in order to achieve a satisfactory behaviour: the proportional term gain  $K_P$ , the gain of the integral term  $K_I$ , and of the derivative one  $K_D$ . These gains are real numbers and characterize the individuals to be evaluated [9, 10].

The objective of the tuning is to minimize the error between the set-point and plant output. Various objective functions were written based on error performance criterion. Each objective function is fundamentally the same except for the section of code that defines the specific error performance criterion being implemented. These object functions are depicted in Tab. 2.

Tab. 2 Parameters of Genetic Algorithm method for calculation the controller parameters.

Criterion	Object function
The Mean Square Error (MSE)	$I_{MSE} = \frac{1}{n} \sum (e(t))^2$
The Integral of Absolute Error (IAE)	$I_{IAE} = \int  e(t)  dt$
The Integral Square Error (ISE)	$I_{ISE} = \int e^2(t) dt$
The Integral of Time multiplied by Absolute Error (ITAE).	$I_{ITAE} = \int t e(t)  dt$

#### 4. Simulation Results

For further investigation the following parameters of the DC motor and converter were used:  $R_a = 3,5 \, \Omega$ ,  $L_a = 0,128 \, \text{H}$ ,  $J_m = 0,15 \, \text{kg.m}^2$ ,  $K_b = 2,7 \, \text{Vs}$ ,  $K_\omega = 0,0666 \, \text{Vs/rad}$ ,  $K_c = 0,4 \, \text{V/A}$ ,  $K_{im} = 120 \, \text{V/V}$ ,  $T_{im} = 0,005 \, \text{s}$ , where the parameters  $K_\omega$  and  $K_c$  are gains of the speed and current sensors.

We calculate the values  $K_a = 1/R_a = 0,28 \, \Omega^{-1}$  and  $T_a = L_a/R_a = 0,0367 \, \text{s}$ .

We developed a program in MATLAB environment for automatic calculation of parameters of the controllers. After numerous tests with different GA parameters such as Population size, Number of generations, and Fitness function for both the speed and cascade controllers we have used MSE as an objective function. The GA parameters are shown in Tab. 2. In both cases of the speed and cascade controller we have used a population size of 60, number of generations equal to 120, and MSE as the object function.

As we can see in Fig. 3 and Fig. 4 the system response comes from transfer function of the system (blue colour) and from the Simulink model of the system (red colour), which means, the MATLAB program deals with the transfer function (TF) and the Simulink model at the same time - by other words: when the user choose system parameters and hits the start button, the MATLAB program calculates the TF of the system, draws its step response, and implements the chosen parameters (parameters of the controllers) into the Simulink model and then draws time response(s) from Simulink.

Tab. 2 Parameters of Genetic Algorithm method for calculation the controller parameters.

GA property	Value/method
Population size	60
Number of generations	120
Fitness function	MSE/ITAE/IAE/ISE
Selection Method	Normalized Geometric selection
Probability of selection	0.05
Crossover method	Arithmetic crossover
Crossover points	4
Mutation method	Uniform mutation
Mutation probability	0,2

Firstly we verified the design method on the model of a drive that is equipped only with one PID speed controller (omitting the current controller) which is the case of small drives only. The controller was optimized by GA method and the correspondent time responses are shown in Fig. 3. The speed response is slower than in the second case to be discussed below and the load torque causes considerable decrease of the speed.

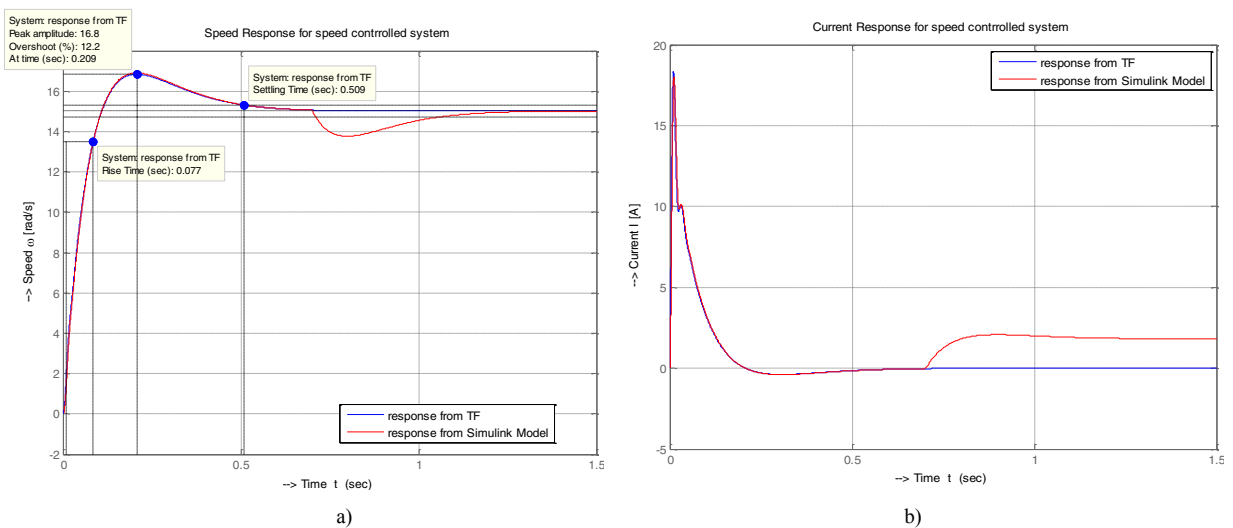


Fig. 3 Time responses for the speed controlled system (the control system without current controller): a) speed, b) current.

Fig. 4a shows speed response of the system with the current and speed controllers in cascade. As it follows from these figures the system response is improved when cascade controller is used in comparison with the system having the speed controlled only (i. e., the overshoot is reduced, and rise time and settling time is increased).

Fig. 4b shows the current response of the drive with current and speed controllers in cascade. As we can observe we have obtained better time responses than in the precious case (faster responses, more robust drive).

Experimentally, using various criteria, we have checked that using the Mean Square Error (MSE) object function gives better results compared with IAE, ITAE, and ISE object functions.

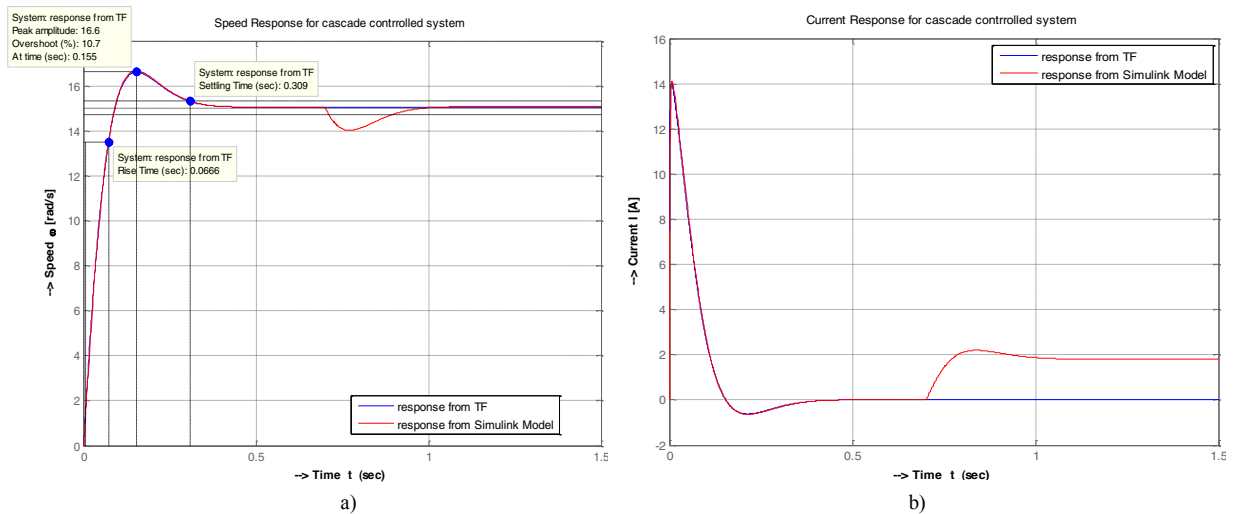


Fig. 4 Time responses for the system with the current and speed controllers connected in cascade: a) speed, b) current.

## 5. Graphical User Interface for Effective Design of Controllers by GAs method

The graphical user interface (GUI) allows user to perform interactively complex tasks through easy changing control elements such as buttons and sliders and to visualize graphs plots, etc., without any knowledge of a mathematical model working in background of the GUI. For effective and easy design of controllers by GA method we have developed the GUI in the MATLAB program – in the GUI Development Environment (GUIDE) that provides tools for designing and programming GUIs, [12, 13].

The screen of the designed GUI for optimisation PI cascade controllers using GAs is shown in Fig. 5.

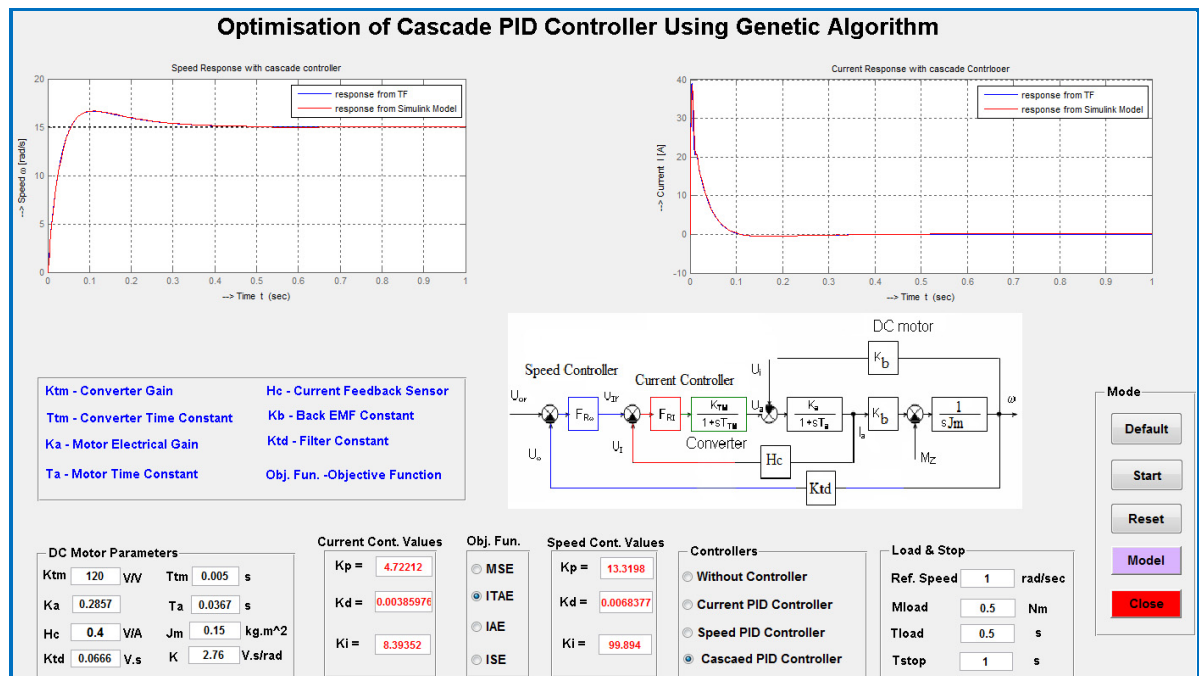


Fig. 5 The complete graphical user interface for design of cascade controllers of the DC drive by the GA method.

The purpose-oriented GUI allows the user:

- To choose the objective function for GA
- To choose the controller
- To change system parameters
- To choose the simulation and load times
- To display the time responses of the system
- To display Simulink model of the system

As shown in Fig. 5, the designed GUI consists of three main parts:

1) *The input part*

The first part consists of four sections: the first one contains system parameters (*DC motor parameters*). The user can implement his system parameters by typing them into their boxes. The second section deals with the calculated parameters of the controllers (*Current Contr. Values, Speed Cont. Values*), the choice of the object function (*Obj. Fun.*) which is chosen by the user simply by clicking the appropriate one. Further there is the section *Controllers* enabling to choose mode of the required system arrangement by clicking the chosen button. In case *Without controller*, there runs simulation of the original system (DC motor) only. The choice *Cascade PID Controller* means, the drive contains both the speed and the current controllers. The section three is the *Load & Stop* section. Here the reference speed, simulation time motor load, and its time are chosen by the user simply by typing them in their boxes. Finally by clicking the *Start* button in the *Mode* section the response of the chosen system with its block diagram is displayed.

2) *The output or displayed part.*

In the upper part of the GUI screen two separate graphs for the time responses of motor speed and current are displayed.

3) *The part for control of GUI modes*

In the *Mode* section there are four buttons: *Default, Reset, Model* and *Close* buttons. When the *Default* button is clicked, all system parameters, values (including the load torque), are returned to default values that are displayed at the first run of the GUI, as well as both axes and the block diagram are deleted. On the other hand, by clicking the *Reset* button, all system parameters and the load & stop values are set to zero. After clicking the *Model* button the Simulink model of the chosen system arrangement is opened in a new window. Finally to close the GUI, just click the *Close* button.

The work with the developed GUI for designing the PID controller parameters is easy – it starts with inputting the plant parameters, choice of the objective function, values of the reference speed, load torque, time of the load and time of simulation (displaying the graphs). According to the chosen controller type the GUI calculates and displays appropriate parameters of the PID controllers and show the time responses of the speed and current.

## Conclusions

Genetic algorithm method of designing the controllers is enough arduous and demanding for the theoretical knowledge of the designer. The developed graphical user interface for design the controllers by GA method considerably facilitate the design task. Here we have used the GUI for designing cascade controllers (the speed and current ones) for the DC drive. The same object function and the same GA parameters can be used for tuning and optimizing both controllers. The simulation results have proved, the cascade controllers give better dynamics of the drive both in the current and speed responses at motor starting and loading (smaller overshoots and faster response). The best simulation results were obtained using the Mean Square Error (MSE) object function (in comparison with the IAE, ITAE, and ISE object criterion functions).

The developed graphical user interface speeds-up the computer-aided designing process of the controllers using GA and enable us also to focus our future research to more complex system with nonlinearities and changeable parameters.

## Acknowledgement



This paper was developed with support of Research and development operating program for the project “University Science Park Technicom for innovative applications with support of knowledge technologies“, code ITMS: 26220220182, cofinanced from European funds.

The work was also supported by the Slovak Cultural and Educational Agency of the Ministry of Education of Slovak Republic under the contract KEGA 042TUKE-4/2012 “Teaching Innovation in Control of Mechatronic Systems”.

## References

- [1] Santana J., Naredo J.L., Sandoval F., Grout I., Argueta O.J.: Simulation and Construction of Speed Control for a DC Series Motor. *Mechatronics*, vol. 12, 2002, pp. 1145-1156.
- [2] Kubin, J.; Richter, A., "Efficiency of mechanical energy recovery from a tram by different input conditions," *15th International Power Electronics and Motion Control Conference (EPE/PEMC), 2012*, vol., no., pp.DS1c.6-1,DS1c.6-5, 4-6 Sept. 2012.
- [3] Saab S.S., Kaed-Bey R.A.: Parameter Identification of a DC Motor: An Experimental Approach. *IEEE International Conf. on Elec. Circuit and Systems (ICECS)*, 2001, pp. 981-984.
- [4] Abdulrahman A.A. Emhemed and Rosbi Bin Mamat: Modelling and Simulation for Industrial DC Motor Using Intelligent Control. *Procedia Engineering*, pp. 420-425. 2012.
- [5] Yildiz A. B.: Electrical equivalent circuit based modeling and analysis of direct current motors. *Electrical Power and Energy Systems*, vol. 43, 2012, pp. 1043–1047.
- [6] Bindu R., Mini K. Namboothiripad: Tuning of PID Controller for DC Servo Motor using Genetic Algorithm. *International Journal of Emerging Technology and Advanced Engineering*, Vol. 2, Issue 3, March 2012, pp. 310-314.
- [7] Tang K. S., Man K. F., Kwong S., He Q.: Genetic Algorithms and their Applications. *IEEE Signal Processing Magazine*. Nov. 1996, pp. 22-37.
- [8] Man K. F., Tang K. S. Kwong S.: Genetic Algorithms: Concepts and Applications. *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, Oct. 1996, pp. 519-534.
- [9] Goldberg. D. E.: *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, 1989.
- [10] Whitley A.: *Genetic Algorithm Tutorial*. Technical Report CS-93-103, Dept. of Computer Science, Colorado State University, 1993.
- [11] Krisztián Lamár, József Neszveda, Average Probability of Failure of Aperiodically Operated Devices, *Acta Polytechnica Hungarica*, Vol.10. No.8. pp.153-167. 2013. – ISSN 1785-8860
- [12] Ismeal G. A. Fedak, V.: Overview of Control Algorithms for DC Motor Drive as a Basis for Development of GUI in MATLAB. *XXVI. microCAD International Scientific Conference, Miskolc 2012*, 9 p., ISBN 978-963-661-773-8.
- [13] Chapman, S. J., *MATLAB Programming for Engineers*. Brooks, 2001.